

(19) 日本国特許庁(JP)

## (12) 公開特許公報(A)

(11) 特許出願公開番号

特開2004-54834

(P2004-54834A)

(43) 公開日 平成16年2月19日(2004. 2. 19)

(51) Int.C1.<sup>7</sup>G06F 1/00  
G09C 1/00

F 1

G06F 9/06  
G09C 1/00

テーマコード(参考)

5B076  
5J104

審査請求 未請求 請求項の数 16 O L (全 22 頁)

(21) 出願番号  
(22) 出願日特願2002-215096 (P2002-215096)  
平成14年7月24日 (2002. 7. 24)(71) 出願人 000005821  
松下電器産業株式会社  
大阪府門真市大字門真1006番地  
(74) 代理人 100077931  
弁理士 前田 弘  
(74) 代理人 100094134  
弁理士 小山 廣毅  
(74) 代理人 100110939  
弁理士 竹内 宏  
(74) 代理人 100110940  
弁理士 嶋田 高久  
(74) 代理人 100113262  
弁理士 竹内 祐二  
(74) 代理人 100115059  
弁理士 今江 克実

最終頁に続く

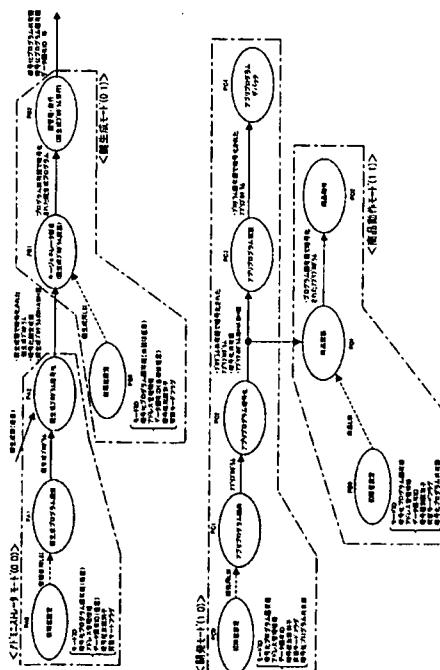
(54) 【発明の名称】 プログラム開発方法、プログラム開発支援装置およびプログラム実装方法

## (57) 【要約】

【課題】 鍵実装システムについて、セキュリティレベルの高い開発環境を提供する。

【解決手段】 セキュアメモリを備えたLSIを有するシステムについて、構成が共通のLSIを、&lt;商品動作モード&gt;とは異なる&lt;開発モード&gt;に設定して、プログラムの開発を行う。また、&lt;アドミニストレータモード&gt;に設定して鍵生成プログラムの開発および暗号化を行い、&lt;鍵生成モード&gt;に設定して、暗号化された鍵生成プログラムを実行させて各種の鍵を生成する。

【選択図】 図2



**【特許請求の範囲】****【請求項 1】**

書き換え不可領域を含むセキュアメモリを備えたLSIを有するシステムに、実装されるプログラムを開発する方法であって、  
前記LSIと構成が共通のLSIを、開発用LSIとして、プログラム実装および製品動作時における商品動作モードとは異なる、開発モードに設定する工程と、  
前記開発用LSIにおいて、前記プログラムの開発を行う工程と  
を備えたことを特徴とするプログラム開発方法。

**【請求項 2】**

請求項1において、  
前記LSIは、  
開発モードに設定されたときは、平文プログラムを実行することができる一方、商品動作モードに設定されたときは、平文プログラムを実行することができないように、その動作が制限される  
ことを特徴とするプログラム開発方法。

10

**【請求項 3】**  
請求項1において、  
前記開発用LSIにおいて、前記プログラム開発工程で開発されたプログラムを、暗号化する暗号化工程を備えた  
ことを特徴とするプログラム開発方法。

20

**【請求項 4】**  
請求項1において、  
前記LSIは、  
開発モードに設定されたとき、平文プログラムを暗号化するための鍵が生成できないよう  
に、その動作が制限される  
ことを特徴とするプログラム開発方法。**【請求項 5】**  
請求項1において、  
前記LSIと構成が共通のLSIを、鍵生成用LSIとして、開発モードおよび実装モードとは異なる鍵生成モードに設定する工程と、  
前記鍵生成用LSIに、暗号化された鍵生成プログラムを実装し、この鍵生成プログラムを実行させることによって、鍵を生成する工程とを備えた  
ことを特徴とするプログラム開発方法。

30

**【請求項 6】**  
請求項5において、  
前記LSIは、  
鍵生成モードに設定されたとき、平文プログラムを実行することができないように、その動作が制限される  
ことを特徴とするプログラム開発方法。**【請求項 7】**  
請求項5において、  
前記LSIと構成が共通のLSIを、管理者用LSIとして、開発モード、実装モードおよび鍵生成モードとは異なるアドミニストレータモードに設定する工程と、  
前記管理者用LSIにおいて、前記鍵生成プログラムを開発し、任意の鍵で暗号化する工程とを備えた  
ことを特徴とするプログラム開発方法。

40

**【請求項 8】**  
暗号化プログラムの開発を支援するプログラム開発支援装置であって、  
前記暗号化プログラムが動作するLSIと構成が共通のLSIと、  
平文プログラムを格納する外部メモリとを備え、

50

前記 L S I は、

生共有鍵に係る共有鍵鍵情報を格納したセキュアメモリを備え、かつ、

前記セキュアメモリに格納された共有鍵鍵情報から、生共有鍵を得る第 1 のステップと、前記外部メモリから入力された平文プログラムを、前記生共有鍵を用いて暗号化する第 2 のステップとが実行可能に構成されている

ことを特徴とするプログラム開発支援装置。

【請求項 9】

暗号化プログラムの開発を支援するプログラム開発支援装置であって、

L S I と、平文プログラムを格納する外部メモリとを備え、

前記 L S I は、

10

生共有鍵に係る共有鍵鍵情報を格納したセキュアメモリと、

ブートプログラムを格納するブート R O M とを備え、かつ、

前記ブート R O M に格納されたブートプログラムを実行することによって、

前記セキュアメモリに格納された共有鍵鍵情報から、生共有鍵を得る第 1 のステップと、前記外部メモリから入力された平文プログラムを、前記生共有鍵を用いて暗号化する第 2 のステップとを実行する

ことを特徴とするプログラム開発支援装置。

【請求項 10】

請求項 8 または 9 において、

前記共有鍵鍵情報は、生共有鍵を生第 1 中間鍵で暗号化した暗号化共有鍵と、前記生第 1 中間鍵を生第 2 中間鍵で暗号化した暗号化第 1 中間鍵とを含むものであり、

20

前記第 1 のステップは、前記暗号化共有鍵および暗号化第 1 中間鍵と、プログラム暗号種とを用いて、前記生共有鍵を復号するものである

ことを特徴とするプログラム開発支援装置。

【請求項 11】

セキュアメモリを有する L S I と、外部メモリとを有する鍵実装システムに、暗号化プログラムを実装する方法であって、

前記セキュアメモリに、生共有鍵に係る共有鍵鍵情報と、生固有鍵に係る固有鍵鍵情報を格納する初期値設定処理と、

前記 L S I において、前記セキュアメモリに格納された共有鍵鍵情報から、生共有鍵を得る第 1 のステップと、

30

前記 L S I において、前記外部メモリから与えられた共有鍵暗号化プログラムを、前記第 1 のステップで得られた生共有鍵を用いて、復号する第 2 のステップと、

前記 L S I において、前記セキュアメモリに格納された固有鍵鍵情報から、生固有鍵を得る第 3 のステップと、

前記 L S I において、前記第 2 のステップで得られた平文プログラムを、前記第 3 のステップで得られた生固有鍵を用いて、暗号化する第 4 のステップとを備え、

前記第 4 のステップで得られた固有鍵暗号化プログラムを、前記外部メモリに実装することを特徴とするプログラム実装方法。

【請求項 12】

40

請求項 11 において、

前記 L S I は、ブートプログラムを格納するブート R O M を備え、

前記ブート R O M に格納されたブートプログラムを前記 L S I に実行させることによって、前記第 1 ～第 4 のステップを実行する

ことを特徴とするプログラム実装方法。

【請求項 13】

請求項 11 において、

前記固有鍵鍵情報は、前記セキュアメモリの書き換え不可領域に、格納されている

ことを特徴とするプログラム実装方法。

【請求項 14】

50

請求項 1 1 において、

前記共有鍵鍵情報は、生共有鍵を生第1中間鍵で暗号化した暗号化共有鍵と、前記生第1中間鍵を生第2中間鍵で暗号化した暗号化第1中間鍵とを含むものであり、

前記第1のステップは、前記暗号化共有鍵および暗号化第1中間鍵と、プログラム暗号種とを用いて、前記生共有鍵を復号するものであることを特徴とするプログラム実装方法。

【請求項 1 5】

請求項 1 1 において、

前記固有鍵鍵情報は、生固有鍵を生第1中間鍵で暗号化した暗号化固有鍵と、前記生第1中間鍵を生第2中間鍵で暗号化した暗号化第1中間鍵とを含むものであり、

10

前記第3のステップは、前記暗号化固有鍵および暗号化第1中間鍵と、プログラム暗号種とを用いて、前記生固有鍵を復号するものであることを特徴とするプログラム実装方法。

【請求項 1 6】

請求項 1 1 において、

前記固有鍵鍵情報は、当該 L S I に固有の固有 I D である

ことを特徴とするプログラム実装方法。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

20

本発明は、鍵実装されたシステムやこれに用いる L S I のプログラム開発やプログラム実装に関する技術に属する。

【0 0 0 2】

【従来の技術】

特願 2 0 0 1 - 2 8 6 8 8 1 では、鍵実装システムにおいて、鍵の機密性および秘匿性を、従来よりも向上させる技術が示されている。

【0 0 0 3】

【発明が解決しようとする課題】

上述のようなシステムでは、そのプログラム開発工程や実装工程においても、いかにしてセキュリティを維持するか、ということが大きな課題となる。

30

【0 0 0 4】

本発明は、上述のようなシステムについて、セキュリティレベルの高い、プログラム開発の方法や環境、またはプログラム実装の方法を提案するものである。

【0 0 0 5】

【課題を解決するための手段】

請求項 1 の発明が講じた解決手段は、書き換え不可領域を含むセキュアメモリを備えた L S I を有するシステムに実装されるプログラムを開発する方法として、前記 L S I と構成が共通の L S I を、開発用 L S I として、プログラム実装および製品動作時における商品動作モードとは異なる開発モードに設定する工程と、前記開発用 L S I において、前記プログラムの開発を行う工程とを備えたものである。

40

【0 0 0 6】

請求項 1 の発明によると、セキュアメモリを備えた L S I を有するシステムに実装されるプログラムの開発が、この L S I と構成が共通であり、かつ、プログラム実装および製品動作時における商品動作モードとは異なる、開発モードに設定された開発用 L S I において、行われる。すなわち、書き換え不可領域を含むセキュアメモリを備えており、高い秘匿性を持つ L S I を、その動作モードを実装モードから開発モードに変えて、プログラム開発環境として用いることによって、プログラム開発環境におけるセキュリティを、従来よりも高めることができる。

【0 0 0 7】

請求項 2 の発明では、請求項 1 における L S I は、開発モードに設定されたときは平文プ

50

ログラムを実行することができる一方、商品動作モードに設定されたときは、平文プログラムを実行することができないようにその動作が制限されるものとする。

【0008】

請求項3の発明では、前記請求項1において、前記開発用LSIにおいて、前記プログラム開発工程で開発されたプログラムを暗号化する暗号化工程を備えたものとする。

【0009】

請求項4の発明では、前記請求項1におけるLSIは、開発モードに設定されたとき、平文プログラムを暗号化するための鍵が生成できないようにその動作が制限されるものとする。

【0010】

請求項5の発明では、前記請求項1において、前記LSIと構成が共通のLSIを鍵生成用LSIとして開発モードおよび実装モードとは異なる鍵生成モードに設定する工程と、前記鍵生成用LSIに暗号化された鍵生成プログラムを実装し、この鍵生成プログラムを実行させることによって鍵を生成する工程とを備えたものとする。

10

【0011】

請求項6の発明では、前記請求項5におけるLSIは、鍵生成モードに設定されたとき、平文プログラムを実行することができないようにその動作が制限されるものとする。

【0012】

請求項7の発明では、前記請求項5において、前記LSIと構成が共通のLSIを管理者用LSIとして、開発モード、実装モードおよび鍵生成モードとは異なるアドミニストレータモードに設定する工程と、前記管理者用LSIにおいて前記鍵生成プログラムを開発し、任意の鍵で暗号化する工程とを備えたものとする。

20

【0013】

請求項8の発明が講じた解決手段は、暗号化プログラムの開発を支援するプログラム開発支援装置として、前記暗号化プログラムが動作するLSIと構成が共通のLSIと、平文プログラムを格納する外部メモリとを備え、前記LSIは、生共有鍵に係る共有鍵鍵情報を格納したセキュアメモリを備え、かつ、前記セキュアメモリに格納された共有鍵鍵情報から生共有鍵を得る第1のステップと、前記外部メモリから入力された平文プログラムを前記生共有鍵を用いて暗号化する第2のステップとが実行可能に構成されているものである。

30

【0014】

請求項8の発明によると、開発対象となる暗号化プログラムが動作するLSIと構成が共通のLSIが、開発環境として、与えられる。そして、このLSIでは、セキュアメモリに格納された共有鍵鍵情報から生共有鍵が得られるとともに、この生共有鍵を用いて、外部メモリから入力された平文プログラムが暗号化される。すなわち、生共有鍵の復号と、この生共有鍵を用いた平文プログラムの暗号化とが、実行可能である。このため、プログラム開発者に生共有鍵を知られることなく、平文プログラムの暗号化を実行することができる。

【0015】

請求項9の発明が講じた解決手段は、暗号化プログラムの開発を支援するプログラム開発支援装置として、LSIと、平文プログラムを格納する外部メモリとを備え、前記LSIは、生共有鍵に係る共有鍵鍵情報を格納したセキュアメモリと、ブートプログラムを格納するブートROMとを備え、かつ、前記ブートROMに格納されたブートプログラムを実行することによって、前記セキュアメモリに格納された共有鍵鍵情報から生共有鍵を得る第1のステップと、前記外部メモリから入力された平文プログラムを前記生共有鍵を用いて暗号化する第2のステップとを実行するものである。

40

【0016】

請求項9の発明によると、LSIにおいてブートプログラムを実行することによって、セキュアメモリに格納された共有鍵鍵情報から生共有鍵が得られるとともに、外部メモリから入力された平文プログラムがこの生共有鍵を用いて暗号化される。すなわち、生共有鍵

50

の復号と、この生共有鍵を用いた平文プログラムの暗号化とが、外部からの指示ではなく、ブートプログラムによって実行される。このため、平文プログラムの暗号化を、プログラム開発者に生共有鍵を知られることを確実に防止しつつ、実行することができる。

【0017】

請求項10の発明では、請求項8または9において、前記共有鍵鍵情報は、生共有鍵を生第1中間鍵で暗号化した暗号化共有鍵と、前記生第1中間鍵を生第2中間鍵で暗号化した暗号化第1中間鍵とを含むものとし、前記第1のステップは、前記暗号化共有鍵および暗号化第1中間鍵と、プログラム暗号種とを用いて、前記生共有鍵を復号するものとする。

【0018】

請求項11の発明が講じた解決手段は、セキュアメモリを有するLSIと外部メモリとを有する鍵実装システムに暗号化プログラムを実装する方法として、前記セキュアメモリに、生共有鍵に係る共有鍵鍵情報と、生固有鍵に係る固有鍵鍵情報とを格納する初期値設定処理と、前記LSIにおいて、前記セキュアメモリに格納された共有鍵鍵情報から生共有鍵を得る第1のステップと、前記LSIにおいて、前記外部メモリから与えられた共有鍵暗号化プログラムを前記第1のステップで得られた生共有鍵を用いて復号する第2のステップと、前記LSIにおいて、前記セキュアメモリに格納された固有鍵鍵情報から生固有鍵を得る第3のステップと、前記LSIにおいて、前記第2のステップで得られた平文プログラムを前記第3のステップで得られた生固有鍵を用いて暗号化する第4のステップとを備え、前記第4のステップで得られた固有鍵暗号化プログラムを前記外部メモリに実装するものである。

10

20

【0019】

請求項11の発明によると、LSIに与えられた共有鍵暗号化プログラムは、セキュアメモリに格納された共有鍵鍵情報から得られた生共有鍵を用いて、復号される。そして復号された平文プログラムは、セキュアメモリに格納された固有鍵鍵情報から得られた生固有鍵を用いて、暗号化される。すなわち、共有鍵暗号化プログラムが、暗号化する鍵が共有鍵から固有鍵に変換されて、システムに実装されることになる。このため、ユーザの持つ各製品では、互いに異なる固有鍵によって暗号化されたプログラムが実装されることになり、秘匿性が向上する。また、万一、暗号を破られた場合でも、被害を受ける製品の数が限定されることになり、従来よりもセキュリティが高まる。

【0020】

30

請求項12の発明では、請求項11におけるLSIは、ブートプログラムを格納するブートROMを備え、前記ブートROMに格納されたブートプログラムを前記LSIに実行させることによって、前記第1～第4のステップを実行するものとする。

【0021】

請求項13の発明では、請求項11における固有鍵鍵情報は、前記セキュアメモリの書き換え不可領域に格納されているものとする。

【0022】

請求項14の発明では、請求項11における共有鍵鍵情報は、生共有鍵を生第1中間鍵で暗号化した暗号化共有鍵と、前記生第1中間鍵を生第2中間鍵で暗号化した暗号化第1中間鍵とを含むものとし、前記第1のステップは、前記暗号化共有鍵および暗号化第1中間鍵とプログラム暗号種とを用いて前記生共有鍵を復号するものとする。

40

【0023】

請求項15の発明では、請求項11における固有鍵鍵情報は、生固有鍵を生第1中間鍵で暗号化した暗号化固有鍵と、前記生第1中間鍵を生第2中間鍵で暗号化した暗号化第1中間鍵とを含むものとし、前記第3のステップは、前記暗号化固有鍵および暗号化第1中間鍵とプログラム暗号種とを用いて前記生固有鍵を復号するものとする。

【0024】

請求項16の発明では、請求項11における固有鍵鍵情報は、当該LSIに固有の固有IDであるものとする。

【0025】

50

## 【発明の実施の形態】

以下、本発明の実施の形態について、図面を参照して説明する。なお、以下の説明では、X（鍵またはプログラム）を鍵Yを用いて暗号化して得た、暗号化された鍵またはプログラムのことを、Enc(X, Y)と表すものとする。

## 【0026】

図1は本実施形態に係るセキュアLSIの内部構成を示すブロック図である。図1において、セキュアLSI1は外部バス120を介して、外部メモリ（フラッシュメモリ）100や外部ツール110などと接続可能に構成されている。また、モードIDを与えることによって、その動作モードを設定することが可能になっている。

## 【0027】

本実施形態に関わる主な構成要素について、簡単に説明する。

10

## 【0028】

まず、セキュアLSI1は、書き換え不可領域11を含むセキュアメモリ（セキュアFlash）10を備えている。この書き換え不可領域11には、書き換え不可領域書き込みフラグ12が設けられている。書き換え不可領域書き込みフラグ12は、モードIDが一度セキュアメモリ10に書き込まれると、そのフラグ値が“可”から“済”になり、それ以降の書き換え不可領域への書き込みが不能になる。なお、本実施形態では、セキュアメモリ10および外部メモリ100はフラッシュメモリによって構成されているが、もちろんこれに限定されるものではなく、不揮発性のメモリであればどのようなものであってもかまわない。

20

## 【0029】

また、秘密鍵演算処理部20は各種の鍵、およびプログラム暗号種を格納するレジスタを備えており、暗号化処理を実行する。鍵生成・更新シーケンサ30はモードID格納レジスタ31を有し、このモードID格納レジスタ31に格納されているモードIDに応じて、秘密鍵演算処理部20の動作、すなわち、各種の鍵を生成できるか否かを制御する。また、鍵またはプログラムがどのようなアルゴリズムや鍵長で暗号化されているかを示す暗号種別識別子を格納する暗号種別識別子格納レジスタ32を備えている。さらに、プログラム暗号種33が実装されている。

## 【0030】

モードシーケンサ40も、モードID格納レジスタ41を備えており、モードID格納レジスタ41に格納されているモードIDと、ジャンパー43の値に応じて、外部ホストインターフェース(I/F)50の動作、すなわち、外部メモリ100に格納されたプログラムやデータをどのI/Fを介して読み込むか、を制御する。これにより、外部メモリ100に格納された平文プログラムが実行できるか否かを制御することができる。さらに、モードシーケンサ40は、鍵がどの手法によって暗号化されているかを示す暗号種別識別子を格納する暗号種別識別子格納レジスタ42を備えている。

30

## 【0031】

外部ホストI/F50は、モードシーケンサ40の制御に従って、プログラム処理部51が有するスルーパート52、遅延部53およびプログラム復号用暗号エンジン54、並びに、データ処理部55が有するスルーパート56およびコンテンツ暗号・復号用暗号エンジン57のうちのいずれかを介して、外部メモリ100や外部ツール110との間でプログラムやデータの入出力を行う。

40

## 【0032】

ここで、後述するアドミニストレータモードを除いては、スルーパート52を介して入力されたプログラムは、セキュアLSI1内部では実行されることはない。すなわち、スルーパート52は、平文プログラムの暗号化、またはすでに暗号化されたプログラムを別の鍵を用いて再暗号化するときに有効とされるものであり、セキュアLSI1は、後述するアドミニストレータモードを除いては、スルーパート52を介して入力されたプログラムへは動作を遷移しないように構成されている。したがって、例えば商品となったセキュアLSI1はスルーパート52を介して平文プログラムを取り込んだとしても、これを実行することはできな

50

い。なお、平文プログラムを実行するときは、セキュアLSI1は遅延部53を介してプログラムをその内部に入力する。

【0033】

ブートROM60は、セキュアLSI1の起動動作を制御するブートプログラムを格納している。HASH演算部70は、セキュアLSI1に読み込まれたプログラムについてその正当性を検証するために、HASH値を演算する。

【0034】

また、外部メモリ100には、プログラムやコンテンツが格納されている。外部ツール110には、セキュアLSI1の最初の起動時にセキュアメモリ10に格納する各種の初期値が格納されている。この初期値の種類は、設定される動作モードに応じて、異なったものになる。

10

【0035】

図2は図1のセキュアLSI1を用いた開発および製品化の全体の流れを表す図である。図2に示すように、セキュアLSI1は、アドミニストレータモード（モードID：00）、鍵生成モード（モードID：01）、開発モード（モードID：10）および商品動作モード（モードID：11）の4種類の動作モードで、動作する。

【0036】

まず、アドミニストレータモードに設定されたセキュアLSI1は、管理者用LSIとして、動作する。管理者用LSIでは、鍵生成プログラムが開発され、また、その鍵生成プログラムが任意の鍵生成鍵を用いて暗号化される。

20

【0037】

鍵生成モードに設定されたセキュアLSI1は、鍵生成用LSIとして、動作する。鍵生成用LSIでは、管理者用LSIにおいて生成された、暗号化された鍵生成プログラムが実装され、この鍵生成プログラムを実行することによって、各種の鍵が生成される。

【0038】

開発モードに設定されたセキュアLSI1は、開発用LSIとして、動作する。開発用LSIでは、実際の製品で実行されるアプリケーション用プログラムが開発される。そして、このアプリケーション用プログラムが、プログラム共有鍵を用いて暗号化される。

【0039】

商品動作モードに設定されたセキュアLSI1は、実際の商品LSIとして、動作する。商品LSIでは、開発用LSIにおいて生成された、プログラム共有鍵で暗号化されたアプリケーション用プログラムが実装され、その内部で、プログラム固有鍵で暗号化されたアプリケーション用プログラムに、変換される。なお、この変換処理は、開発用LSIでも、アプリケーション用プログラムのデバッグのために、実行可能になっている。

30

【0040】

以下、各モードにおけるセキュアLSI1の動作の詳細について、フローチャートおよびデータフローを参照して、説明する。セキュアLSI1は、ブートROM60に格納されたブートプログラムを実行することによって、以下のような動作を行う。

【0041】

図3はブートプログラムの全体的な処理の流れを示すフローチャートである。セキュアLSI1に電源が投入されると、ブートROM60に格納されたブートプログラムがCPU65によって実行される。図3に示すように、まず、各ハードウェアを初期化する(SZ0)。そして、外部ツール110からさまざまな初期値を読み込み、セキュアメモリ10に設定する(SZ1)。

40

【0042】

図22は初期値設定処理SZ1のフローチャートである。まず、ジャンパー44で、セキュアメモリ10がLSI内に実装されているか否かの判定を行う。次に、書き換え不可領域書き込みフラグ12が“済”であるか否かを判定し、“済”であるときはすでにセキュアメモリ10に初期値が設定されているので、処理SZ1を終了する。書き換え不可領域書き込みフラグ12が“可”であるときは、セキュアメモリ10に初期値を書き込んでい

50

く。モードIDに加えて、暗号化されたプログラム固有鍵、アドレス管理情報、データ固有鍵をセキュアメモリ10の書き換え不可領域11に書き込む。なお、最初の判定の結果、セキュアメモリ10がLSIの外部にあると判定されたときは、モードIDは商品動作モードを表す値に上書きされる。これにより、セキュアメモリ10がLSIパッケージ外にあるような不正な製品は、商品動作モードでしか動作できない。

#### 【0043】

次に、書き込み不可領域書き込みフラグ12を“済”にセットする。これによって、以後の書き換え不可領域11の書き換えはできなくなる。さらに、通常領域13、14に暗号種別識別子および実装モードフラグを書き込む。そして、モードIDがアドミニストレータモード以外のモードを示すときは、これらに加えて、暗号化された共有鍵／鍵生成鍵も通常領域13、14に書き込む。

10

#### 【0044】

その後、前処理SZ2を実行する。図4は前処理SZ2のデータフローである。ここでは、セキュアメモリ10の書き込み不可領域11に設定されたモードIDが、鍵生成・更新シーケンサ30のモードID格納レジスタ31と、モードシーケンサ40のモードID格納レジスタ41とに設定される。また、セキュアメモリ10の第1の通常領域13に設定された暗号種別識別子が、鍵生成・更新シーケンサ30の暗号種別識別子格納レジスタ32と、モードシーケンサ40の暗号種別識別子格納レジスタ42とに設定される。さらに、セキュアメモリ10の書き換え不可領域11に格納されたアドレス管理情報が、MEMC80の暗号アドレス区分格納レジスタ81に設定される。ここまで動作は、図2における初期値設定フェーズPA0、PB0、PC0、PD0に対応している。

20

#### 【0045】

その後は、モードIDの値に応じて、それぞれのモードにおける動作を行う(SZ3)。

#### 【0046】

＜アドミニストレータモード＞

モードIDが「00」のとき、セキュアLSI1はアドミニストレータモードになり、ジャンパー43の値に応じて(SA0)、平文プログラム実行処理SA1、またはプログラム暗号化処理SA2を実行する。

#### 【0047】

鍵生成プログラム開発フェーズPA1では、平文プログラム実行処理SA1が行われ、ここで、鍵生成プログラムが生成される。この鍵生成プログラムは外部メモリ100に格納される。

30

#### 【0048】

鍵生成プログラム暗号化フェーズPA2では、まず、図5のデータフローのように、鍵生成プログラムを実行させることによって、与えられた任意の鍵生成鍵を暗号化する。すなわち、外部ホスト1/F50では、モードシーケンサ40によって、プログラム処理部51のスルーパート52が有効化される。そして、外部メモリ100に格納された鍵生成プログラムが、スルーパート52を介してCPU65に与えられ、実行される。この鍵生成プログラムを実行することによって、外部メモリ100に格納された鍵生成鍵が、秘密鍵演算処理部20によって、鍵生成・更新シーケンサ30に実装されたプログラム暗号種を用いて暗号化される。

40

#### 【0049】

なお、本実施形態では、鍵の暗号化は、第1中間鍵と第2中間鍵とを用いて行う。すなわち、暗号化の結果、平文鍵(ここでは鍵生成鍵)を第1中間鍵(ここではMK1)で暗号化した暗号化鍵(ここではEnc(鍵生成鍵、MK1))と、第1中間鍵を第2中間鍵(ここではCK)で暗号化した暗号化第1中間鍵(ここではEnc(MK1, CK))とが得られる。もちろん、本発明は、このような鍵の暗号化手法に限定されるものではない。

#### 【0050】

その後、プログラム暗号化処理SA2が実行される。図6はこのプログラム暗号化処理SA2のフローチャート、図7はデータフローである。まず、外部メモリ100に格納され

50

ている、暗号化された鍵生成鍵 E n c (鍵生成鍵, MK1), E n c (MK1, CK) を、外部ホスト I / F 50 のスルーパート 52 を介して、秘密鍵演算処理部 20 に設定する (S A 21)。そして、この暗号化された鍵生成鍵を、鍵生成・更新シーケンサ 30 に実装されたプログラム暗号種を用いて復号し、鍵生成鍵を得る (S A 22)。その後、外部メモリ 100 に格納されていた平文の鍵生成プログラムを取り込み、これを S A 22 で復号した鍵生成鍵を用いて暗号化し、外部メモリ 100 に書き込む (S A 23)。さらに、外部メモリ 100 の平文の鍵生成プログラムに対して、H A S H 演算部 70 によって H A S H 演算を行い、算出した H A S H 値を外部メモリ 100 に書き込む (S A 24)。

## 【0051】

このような動作によって、アドミニストレータモードでは、鍵生成鍵で暗号化された鍵生成プログラム E n c (鍵生成プログラム、鍵生成鍵) と、暗号化された鍵生成鍵 E n c (鍵生成鍵, MK1), E n c (MK1, CK) と、鍵生成プログラムの H A S H 値とが、生成される。

## 【0052】

## &lt;鍵生成モード&gt;

モード I D が「01」のとき、セキュア L S I 1 は鍵生成モードになり、実装モードフラグの値に応じて (S B 0)、キージェネレータ製造処理 S B 1、または鍵管理・発行処理 S B 2 を実行する。

## 【0053】

キージェネレータ製造フェーズ P B 1 では、キージェネレータ製造処理 S B 1 が実行される。図 8 はこの処理 S B 1 のフローチャート、図 9 および図 10 はデータフローである。ここでは、モード I D と実装モードフラグの値によって、外部ホスト I / F 部 50 が有するプログラム処理部 51 においてスルーパート 52 が有効に設定されている。

## 【0054】

まず、セキュアメモリ 10 の書き込み不可領域 11 に格納されている、暗号化されたプログラム固有鍵 E n c (プログラム固有鍵、MK0), E n c (MK0, CK) を秘密鍵演算処理部 20 の暗号鍵格納レジスタに設定する (S B 11)。そして、この暗号化されたプログラム固有鍵を、鍵生成・更新シーケンサ 30 に実装されたプログラム暗号種を用いて復号し、プログラム固有鍵を得る (S B 12)。次に、初期値設定フェーズ P B 0 において設定された、暗号化された鍵生成鍵 E n c (鍵生成鍵, MK1), E n c (MK1, CK) を秘密鍵演算処理部 20 の暗号鍵格納レジスタに設定し (S B 13)、この暗号化された鍵生成鍵を、鍵生成・更新シーケンサ 30 に実装されたプログラム暗号種を用いて復号し、鍵生成鍵を得る (S B 14)。

## 【0055】

その後、外部メモリ 100 に格納されていた、鍵生成鍵で暗号化された鍵生成プログラム E n c (鍵生成プログラム、鍵生成鍵) を、外部ホスト I / F 50 が有するプログラム処理部 51 のスルーパート 52 を介して、秘密鍵演算処理部 20 に取り込む (S B 15)。そして、取り込んだ暗号化された鍵生成プログラムを、鍵生成鍵で復号した後、プログラム固有鍵で暗号化し、暗号化された鍵生成プログラム E n c (鍵生成プログラム、プログラム固有鍵)を得る (S B 16)。そして、外部メモリ 100 に書き込む (S B 17)。次に、外部メモリ 100 に格納されていた H A S H 値を、スルーパート 52 を介して、セキュアメモリ 10 の通常領域 13 に設定する (S B 18)。

## 【0056】

また、セキュアメモリ 10 の通常領域 13 に格納された実装モードフラグの値を、C P U 65 によって “O F F” に設定する (S B 19)。そして、セキュアメモリ 10 の通常領域 13 に格納されている、暗号化された鍵生成鍵 E n c (鍵生成鍵, MK1), E n c (MK1, CK) を削除する (S B 1A) とともに、外部メモリ 100 に格納されていた、暗号化された鍵生成プログラム E n c (鍵生成鍵プログラム、鍵生成鍵) および H A S H 値を削除する (S B 1B)。

## 【0057】

10

20

30

40

50

鍵管理・発行フェーズ P B 2 では、鍵管理・発行処理 S B 2 が実行される。図 1 1 はこの処理 S B 2 のフローチャート、図 1 2 および図 1 3 はデータフローである。ここでは、モード I D と実装モードフラグの値によって、外部ホスト I / F 部 5 0 が有するプログラム復号用暗号エンジン 5 4 が有効に設定されている。

#### 【0058】

まず、セキュアメモリ 1 0 の書き込み不可領域 1 1 に格納されている、暗号化されたプログラム固有鍵 E n c (プログラム固有鍵、M K 0)、E n c (M K 0, C K) を秘密鍵演算処理部 2 0 の暗号鍵格納レジスタに設定する (S B 2 1)。そして、この暗号化されたプログラム固有鍵を、鍵生成・更新シーケンサ 3 0 に実装されたプログラム暗号種を用いて復号し、プログラム固有鍵を得る (S B 2 2)。得たプログラム固有鍵は、外部ホスト I / F 5 0 のプログラム復号用暗号エンジン 5 4 のプログラム固有鍵格納レジスタに設定される (S B 2 3)。 10

#### 【0059】

その後、外部メモリ 1 0 0 に格納されていた、プログラム固有鍵で暗号化された鍵生成プログラム E n c (鍵生成プログラム、プログラム固有鍵) を、外部ホスト I / F 5 0 が有するプログラム処理部 5 1 のプログラム復号用暗号エンジン 5 4 を介して復号し、H A S H 演算部 7 0 に取り込み、H A S H 値を演算する (S B 2 4)。そして、この演算したH A S H 値と、セキュアメモリ 1 0 の通常領域 1 3 に格納されていたH A S H 値とを比較し、鍵生成プログラムが改ざんされていないかどうかをチェックする (S B 2 5)。H A S H 値が一致していたとき (S B 2 6 で N o)、外部メモリ 1 0 0 に格納されていた鍵生成プログラム E n c (鍵生成プログラム、プログラム固有鍵) に処理を遷移し、鍵の生成を実行する (S B 2 7)。一方、H A S H 値が一致していないとき (S B 2 6 で Y e s) は、何らかの不正が行われたものと推定して、不正アクセス時制御による処理を実行する (S B 2 8)。 20

#### 【0060】

鍵生成モードにおいては、スルーパ 5 2 を有効にしてプログラムを入力する、またはプログラム復号用暗号エンジン 5 4 を有効にして暗号化されたプログラムを復号して入力するのみであるので、平文プログラムを実行することができないように、セキュア L S I 1 の動作が制限される。

#### 【0061】

＜開発モード＞

モード I D が「1 0」のとき、セキュア L S I 1 は開発モードになり、ジャンパー 4 3 の値に応じて (S C 0)、プログラム暗号化処理 S C 1、平文プログラム実行処理 S C 2、プログラム実装処理 S C 3、または暗号化プログラム実行処理 S C 4 を実行する。 30

#### 【0062】

アプリケーションプログラム開発フェーズ P C 1 では、遅延部 5 3 を有効として、平文プログラム実行処理 S C 2 が行われ、アプリケーションプログラムが開発される。開発されたアプリケーションプログラムは、外部メモリ 1 0 0 に格納される。

#### 【0063】

アプリケーションプログラム暗号化フェーズ P C 2 では、プログラム暗号化処理 S C 1 が実行される。図 1 4 はこのプログラム暗号化処理 S C 1 のフローチャート、図 1 5 はデータフローである。まず、セキュアメモリ 1 0 の通常領域 1 4 に格納された共有鍵鍵情報としての暗号化されたプログラム共有鍵 E n c (プログラム共有鍵、M K 2)、E n c (M K 2, C K) を秘密鍵演算処理部 2 0 に設定する (S C 1 1)。そして、この暗号化されたプログラム共有鍵を、鍵生成・更新シーケンサ 3 0 に実装されたプログラム暗号種を用いて復号し、プログラム共有鍵を得る (S C 1 2)。その後、外部メモリ 1 0 0 に格納された平文のアプリケーションプログラムを取り込み、これを S C 1 2 で復号したプログラム共有鍵を用いて暗号化し、外部メモリ 1 0 0 に書き込む (S C 1 3)。さらに、外部メモリ 1 0 0 の平文のアプリケーションプログラムに対して、H A S H 演算部 7 0 によって H A S H 演算を行い、算出したH A S H 値を外部メモリ 1 0 0 に書き込む (S C 1 4)。 40 50

## 【0064】

このような動作によって、プログラム共有鍵で暗号されたアプリケーションプログラム E n c (アプリケーションプログラム、プログラム共有鍵) と、アプリケーションプログラムの H A S H 値とが、生成される。

## 【0065】

次に、アプリケーションプログラム実装フェーズ P C 3 では、プログラム実装処理 S C 3 が実行され、アプリケーションプログラムデバッグフェーズ P C 4 では、暗号化プログラム実行処理 S C 4 が実行される。これらの処理は、商品動作モードにおける各処理 S D 1 , S D 2 と同様であるので、詳細は後述する。

## 【0066】

10 このように、書き換え不可領域 1 1 を含むセキュアメモリ 1 0 を有し、高い秘匿性を持つ L S I 1 を、その動作モードを実装モードから開発モードに変えてプログラム開発環境として用いることによって、プログラム開発環境におけるセキュリティを、従来よりも高めることができる。

## 【0067】

また、セキュアメモリ 1 0 に格納された共有鍵鍵情報としての暗号化された共有鍵から生共有鍵が復号され、そしてこの生共有鍵を用いて平文プログラムの暗号化が実行されるので、プログラム開発者に生共有鍵を知られることなく、平文プログラムの暗号化を実行することができる。

## 【0068】

また、生共有鍵の復号と、この生共有鍵を用いた平文プログラムの暗号化とが、外部からの指示ではなく、ブートプログラムによって実行されるので、平文プログラムの暗号化を、プログラム開発者に生共有鍵を知られることを確実に防止しつつ、実行することができる。

## 【0069】

＜商品動作モード＞

モード I D が「1 1」のとき、セキュア L S I 1 は商品動作モードになり、実装モードフラグの値に応じて ( S D 0 ) 、プログラム実装処理 S D 1 、または通常ブート処理 S D 2 を実行する。

## 【0070】

商品実装フェーズ P D 1 では、プログラム実装処理 S D 1 が実行される。図 1 6 はこの処理 S D 1 のフローチャート、図 1 7 および図 1 8 はデータフローである。ここでは、モード I D と実装モードフラグの値によって、外部ホスト I / F 部 5 0 が有するプログラム処理部 5 1 においてスルーパ 5 2 が有効に設定されている。

## 【0071】

まず、セキュアメモリ 1 0 の書き込み不可領域 1 1 に格納された、固有鍵鍵情報としての暗号化されたプログラム固有鍵 (プログラム固有鍵、M K 0 ) , E n c (M K 0 , C K ) を秘密鍵演算処理部 2 0 の暗号鍵格納レジスタに設定する ( S D 1 1 ) 。そして、この暗号化されたプログラム固有鍵を、鍵生成・更新シーケンサ 3 0 に実装されたプログラム暗号種を用いて復号し、プログラム固有鍵を得る ( S D 1 2 ) 。次に、初期値設定フェーズ P D 0 において設定された、共有鍵鍵情報としての暗号化されたプログラム共有鍵 E n c (プログラム共有鍵、M K 2 ) , E n c (M K 2 , C K ) を秘密鍵演算処理部 2 0 の暗号鍵格納レジスタに設定し ( S D 1 3 ) 、この暗号化されたプログラム共有鍵を、鍵生成・更新シーケンサ 3 0 に実装されたプログラム暗号種を用いて復号し、プログラム共有鍵を得る ( S D 1 4 ) 。

## 【0072】

その後、外部メモリ 1 0 0 に格納されていた、プログラム共有鍵で暗号化されたアプリケーションプログラム E n c (アプリケーションプログラム、プログラム共有鍵) を、外部ホスト I / F 5 0 が有するプログラム処理部 5 1 のスルーパ 5 2 を介して、秘密鍵演算処理部 2 0 に取り込む ( S D 1 5 ) 。そして、取り込んだ暗号化されたアプリケーションプ

10

20

30

40

50

ログラムを、プログラム共有鍵で復号した後、プログラム固有鍵で暗号化し、暗号化されたアプリケーションプログラム E n c (アプリケーションプログラム、プログラム固有鍵)を得る (S D 1 6)。そして、外部メモリ 1 0 0 に書き込む (S D 1 7)。次に、外部メモリ 1 0 0 に格納されていた H A S H 値を、スルーパ 5 2 を介して、セキュアメモリ 1 0 の通常領域 1 3 に設定する (S D 1 8)。

【0 0 7 3】

また、セキュアメモリ 1 0 の通常領域 1 3 に格納された実装モードフラグの値を、C P U 6 5 によって “O F F” に設定する (S D 1 9)。そして、セキュアメモリ 1 0 の通常領域 1 3 に格納されている、暗号化されたプログラム共有鍵 E n c (プログラム共有鍵, M K 1), E n c (M K 1, C K) を削除する (S D 1 A) とともに、外部メモリ 1 0 0 に格納されていた、暗号化されたアプリケーションプログラム E n c (アプリケーションプログラム、プログラム共有鍵) および H A S H 値を削除する (S D 1 B)。

10

【0 0 7 4】

すなわち、共有鍵暗号化プログラムが、暗号化する鍵が共有鍵から固有鍵に変換されて、システムに実装されることになる。このため、ユーザの持つ各製品では、互いに異なる固有鍵によって暗号化されたプログラムが実装されることになり、秘匿性が向上する。また、万一、暗号を破られた場合でも、被害を受ける製品の数が限定されることになり、従来よりもセキュリティが高まる。

【0 0 7 5】

なお、固有鍵の生成は、固有 I D を基にして行ってもよい。すなわち、セキュア L S I 1 每に、個別の固有 I D を固有鍵鍵情報としてセキュアメモリ 1 0 に実装しておき、この商品実装フェーズ P D 1 において、ブートプログラムによって、実装された固有 I D から固有鍵を生成するようにしてもよい。

20

【0 0 7 6】

商品動作フェーズ P D 2 では、通常ブート処理 S D 2 が実行される。図 1 9 はこの処理 S D 2 のフローチャート、図 2 0 および図 2 1 はデータフローである。ここでは、モード I D と実装モードフラグの値によって、外部ホスト I / F 部 5 0 が有するプログラム復号用暗号エンジン 5 4 が有効に設定されている。

【0 0 7 7】

まず、セキュアメモリ 1 0 の書き込み不可領域 1 1 に格納されている、暗号化されたプログラム固有鍵 E n c (プログラム固有鍵、M K 0), E n c (M K 0, C K) を秘密鍵演算処理部 2 0 の暗号鍵格納レジスタに設定する (S D 2 1)。そして、この暗号化されたプログラム固有鍵を、鍵生成・更新シーケンサ 3 0 に実装されたプログラム暗号種を用いて復号し、プログラム固有鍵を得る (S D 2 2)。得たプログラム固有鍵は、外部ホスト I / F 5 0 のプログラム復号用暗号エンジン 5 4 のプログラム固有鍵格納レジスタに設定する (S D 2 3)。

30

【0 0 7 8】

その後、セキュアメモリ 1 0 の書き込み不可領域 1 1 に格納されているデータ固有 I D を秘密鍵演算処理部 2 0 の固有 I D 格納レジスタに設定する (S D 2 4)。また、C P U 6 5 によって乱数を生成し、秘密鍵演算処理部 2 0 の乱数格納レジスタに設定する (S D 2 5)。そして、秘密鍵演算処理部 2 0 によって、データ固有 I D と乱数からデータ固有鍵を生成する (S D 2 6)。

40

【0 0 7 9】

その後、外部メモリ 1 0 0 に格納されていた、プログラム固有鍵で暗号化されたアプリケーションプログラム E n c (アプリケーションプログラム、プログラム固有鍵) を、外部ホスト I / F 5 0 が有するプログラム処理部 5 1 のプログラム復号用暗号エンジン 5 4 を介して復号し、H A S H 演算部 7 0 に取り込み、H A S H 値を演算する (S D 2 7)。そして、この演算した H A S H 値と、セキュアメモリ 1 0 の通常領域 1 3 に格納されていた H A S H 値とを比較し、アプリケーションプログラムが改ざんされていないかどうかをチェックする (S D 2 8)。H A S H 値が一致していたとき (S D 2 9 で N o) 、外部メモ

50

リ 1 0 0 に格納されていたアプリケーションプログラム E n c (アプリケーションプログラム, プログラム固有鍵) に処理を遷移し、アプリケーションを実行する (S D 2 A)。一方、H A S H 値が一致していないとき (S D 2 9 で Y e s) は、何らかの不正が行われたものと推定して、不正アクセス時制御による処理を実行する (S D 2 B)。

【0 0 8 0】

商品動作モードにおいては、スルーポート 5 2 を有効にしてプログラムを入力する、またはプログラム復号用暗号エンジン 5 4 を有効にして暗号化されたプログラムを復号して入力するのみであるので、平文プログラムを実行することができないように、セキュア L S I 1 の動作が制限される。

【0 0 8 1】

なお、開発モードおよび商品動作モードにおいては、外部から、秘密鍵演算処理部 2 0 を用いて鍵を生成する処理を実行させようとしても、鍵生成・更新シーケンサ 3 0 によって判別されて実行されない。すなわち、鍵生成・更新シーケンサ 3 0 は、開発モードおよび商品動作モードにおいては、起動時以外にプログラム暗号種を用いることができないように動作を制限するので、鍵を生成する処理は実行することができない。

10

【0 0 8 2】

なお、本実施形態では、外部メモリ 1 0 0 にプログラムやデータが格納されており、外部ツール 1 1 0 に、セキュアメモリ 1 0 に実装される初期値が格納されているが、これらはどちらに格納されていてもかまわない。例えば、プログラムやデータが外部ツール 1 1 0 から読み込まれ、再暗号化されたとしても何ら問題はない。

20

【0 0 8 3】

なお、本実施形態では、ブートプログラムによって、各処理を実行するものとしたが、本発明はこれに限られるものではなく、処理の一部または全部を、他の手段によって実行してもかまわない。ただし、外部からの指示ではなく、ブートプログラムによって処理を実行させることによって、セキュリティをより高めることができる。

【0 0 8 4】

【発明の効果】

以上のように本発明によると、書き換え不可領域を含むセキュアメモリを有し、高い秘匿性を持つ L S I を、その動作モードを実装モードから開発モードに変えて、プログラム開発環境として用いることによって、プログラム開発環境におけるセキュリティを、従来よりも高めることができる。

30

【図面の簡単な説明】

【図 1】本発明の実施形態に係るセキュア L S I の構成を示すブロック図である。

【図 2】図 1 のセキュア L S I を用いた開発および製品化の全体の流れを表す図である。

【図 3】ブートプログラムの全体的な処理の流れを示すフローチャートである。

【図 4】前処理 S Z 2 のデータフローである。

【図 5】鍵生成鍵の暗号化のデータフローである。

【図 6】プログラム暗号化処理 S A 2 のフローチャートである。

【図 7】プログラム暗号化処理 S A 2 のデータフローである。

【図 8】鍵生成モードにおけるキージェネレータ製造処理 S B 1 のフローチャートである

40

。【図 9】キージェネレータ製造処理 S B 1 のデータフローである。

【図 1 0】キージェネレータ製造処理 S B 1 のデータフローである。

【図 1 1】鍵生成モードにおける鍵管理・発行処理 S B 2 のフローチャートである。

【図 1 2】鍵管理・発行処理 S B 2 のデータフローである。

【図 1 3】鍵管理・発行処理 S B 2 のデータフローである。

【図 1 4】開発モードにおけるプログラム暗号化処理 S C 1 のフローチャートである。

【図 1 5】プログラム暗号化処理 S C 1 のデータフローである。

【図 1 6】商品動作モードにおけるプログラム実装処理 S D 1 のフローチャートである。

【図 1 7】プログラム実装処理 S D 1 のデータフローである。

50

【図18】プログラム実装処理SD1のデータフローである。

【図19】商品動作モードにおける通常ブート処理SD2のフローチャートである。

【図20】通常ブート処理SD2のデータフローである。

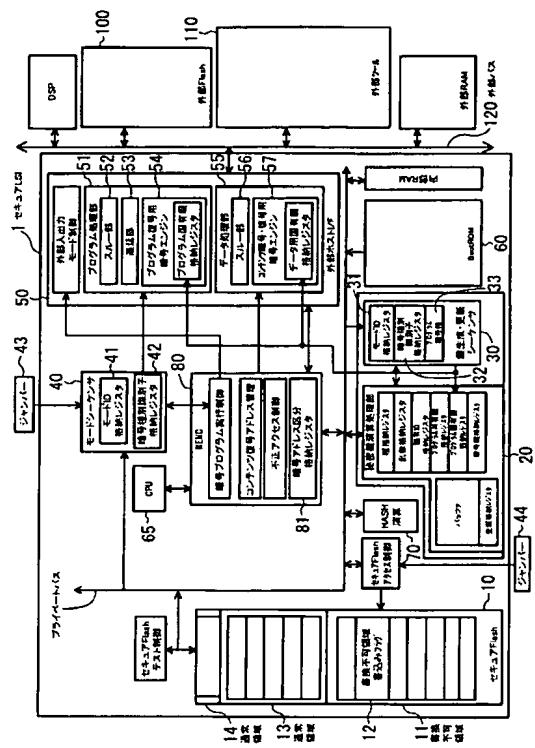
【図21】通常ブート処理SD2のデータフローである。

【図22】初期値設定処理SZ1のフローチャートである。

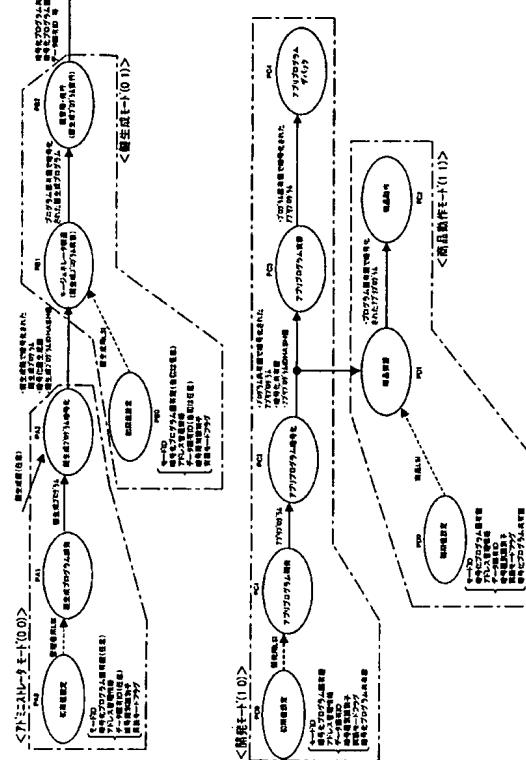
【符号の説明】

1 0	セキュアメモリ	
1 1	書き換え不可領域	
1 2	書き換え不可領域書き込みフラグ	
2 0	秘密鍵演算処理部	10
3 0	鍵生成・更新シーケンサ	
3 1	モードID格納レジスタ	
3 2	暗号種別識別子格納レジスタ	
3 3	プログラム暗号種	
4 0	モードシーケンサ	
4 1	モードID格納レジスタ	
4 2	暗号種別識別子格納レジスタ	
5 0	外部ホストI/F	
5 1	プログラム処理部	
5 2	スルーポート	20
5 3	遅延部	
5 4	プログラム復号用暗号エンジン	
5 5	データ処理部	
5 6	スルーポート	
5 7	コンテンツ暗号・復号用暗号エンジン	
6 0	ブートROM	
7 0	HASH演算部	
1 0 0	外部メモリ	

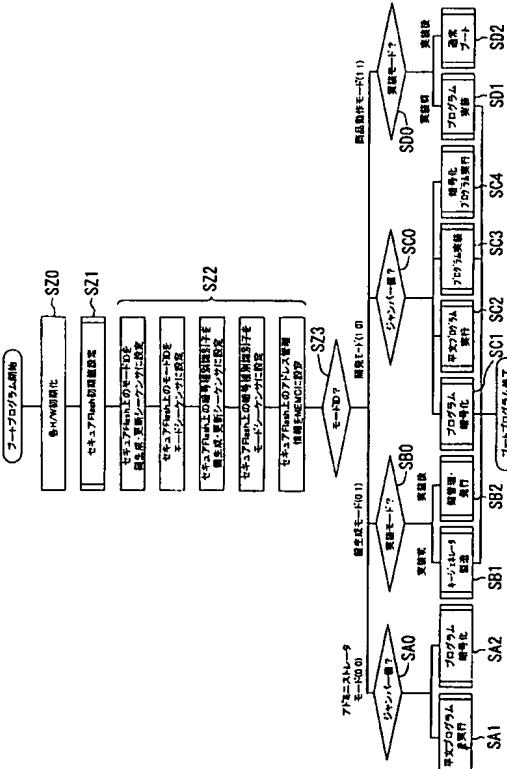
【図 1】



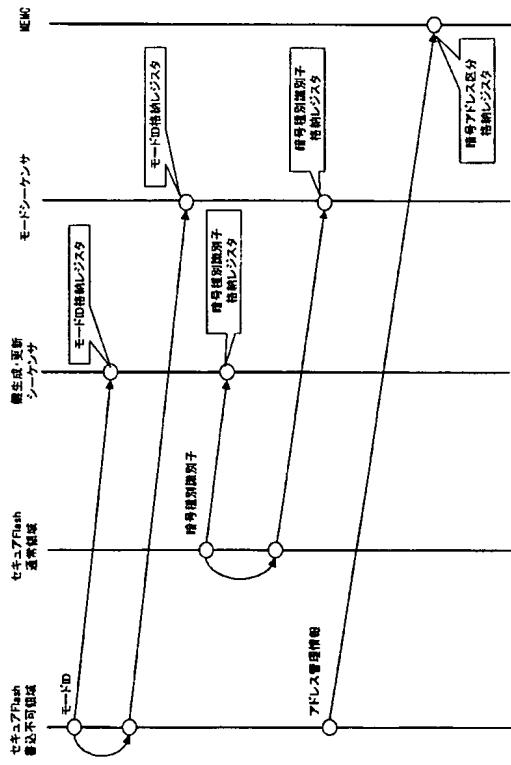
【図2】



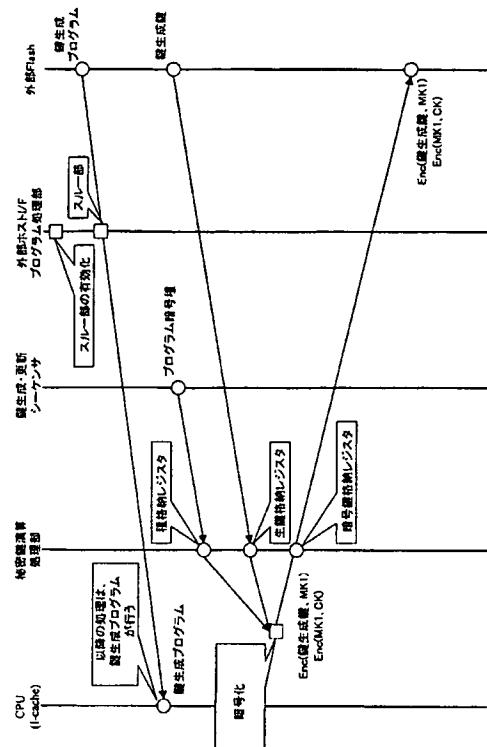
〔 図 3 〕



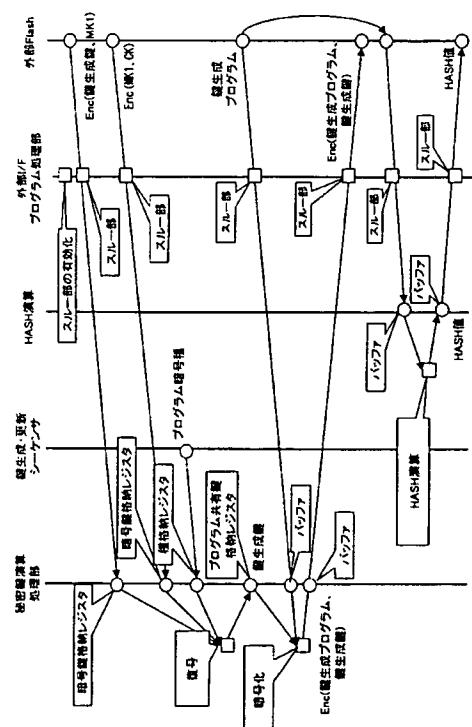
〔四四〕



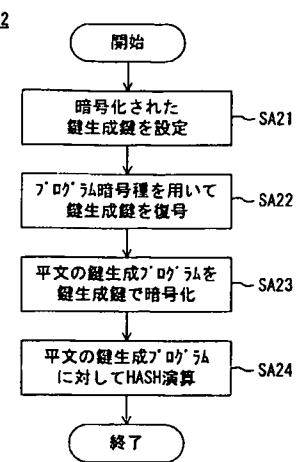
〔図5〕



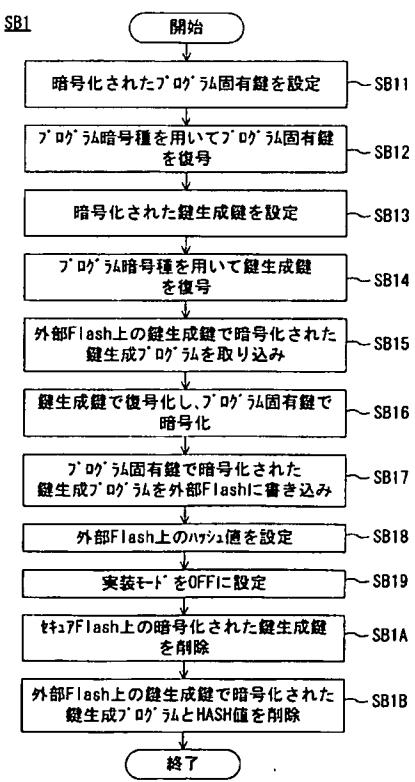
〔図7〕



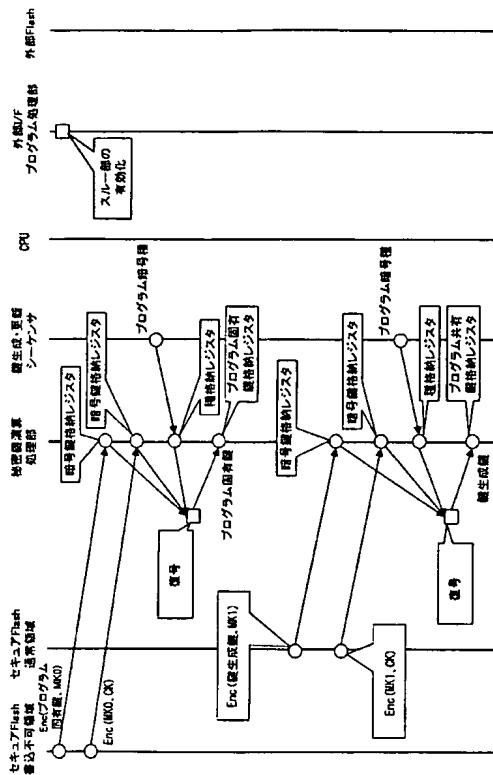
〔図6〕



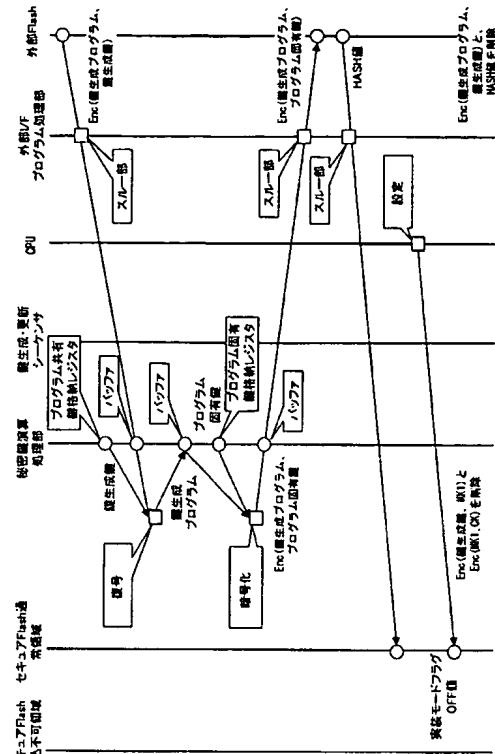
〔図8〕



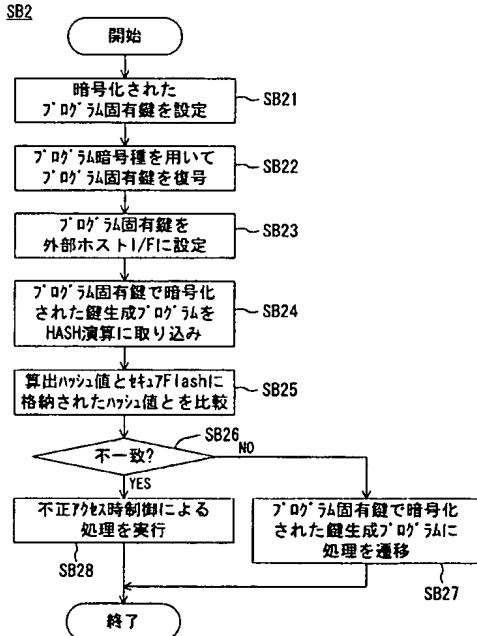
〔図9〕



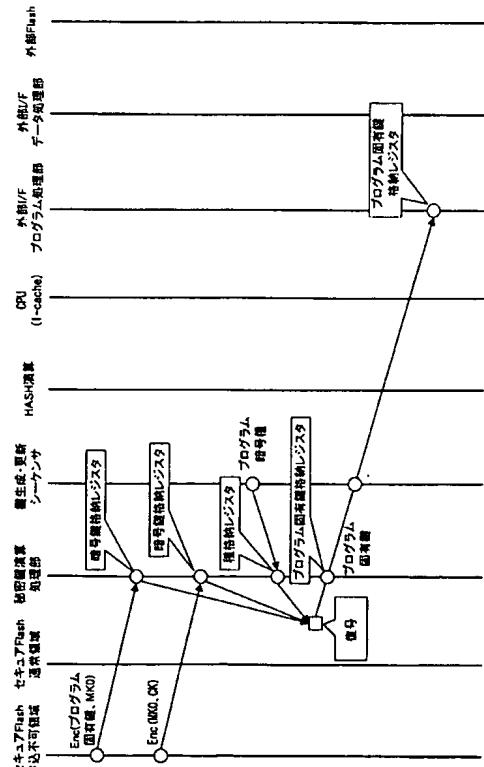
【図 10】



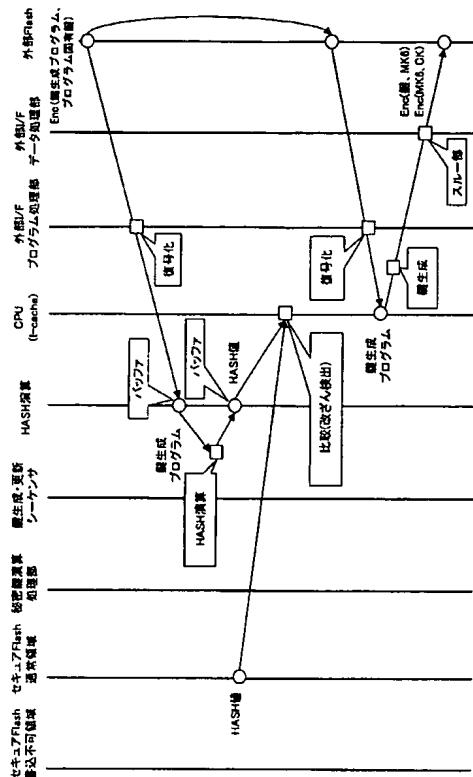
【図 1 1】



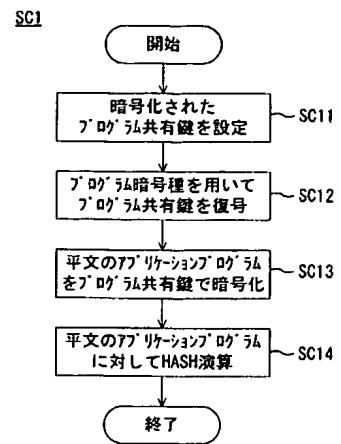
【図 1 2】



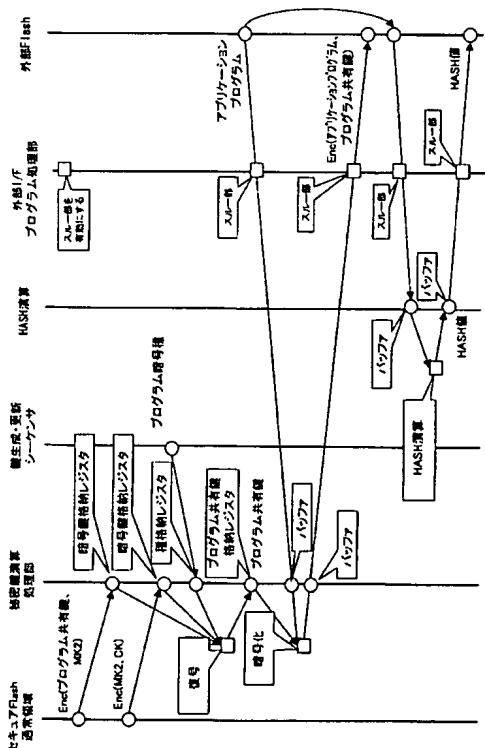
【图 1 3】



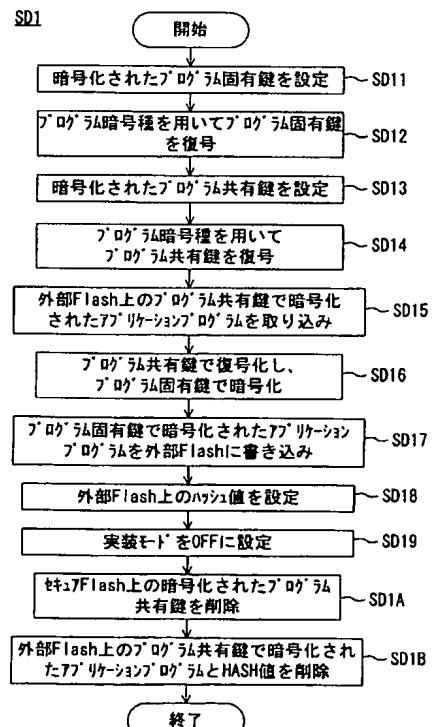
【图 1-4】



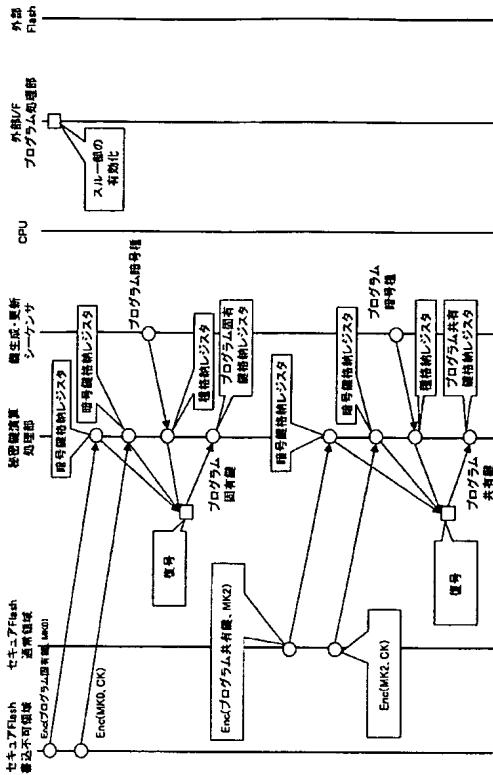
【図 1 5】



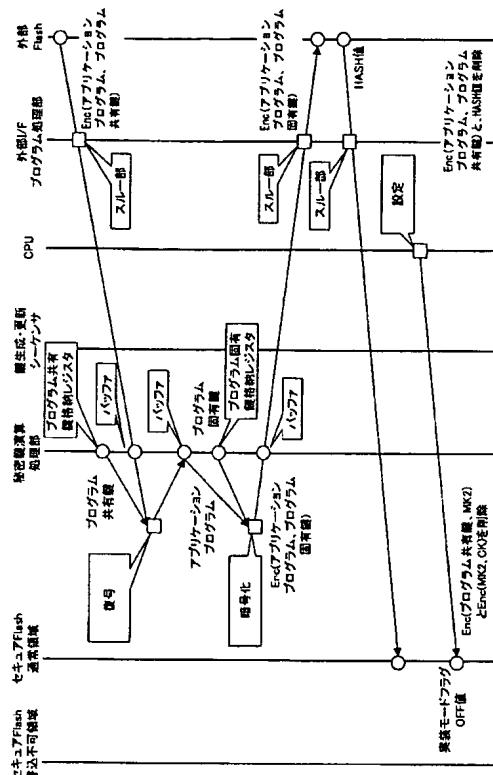
【 16 】



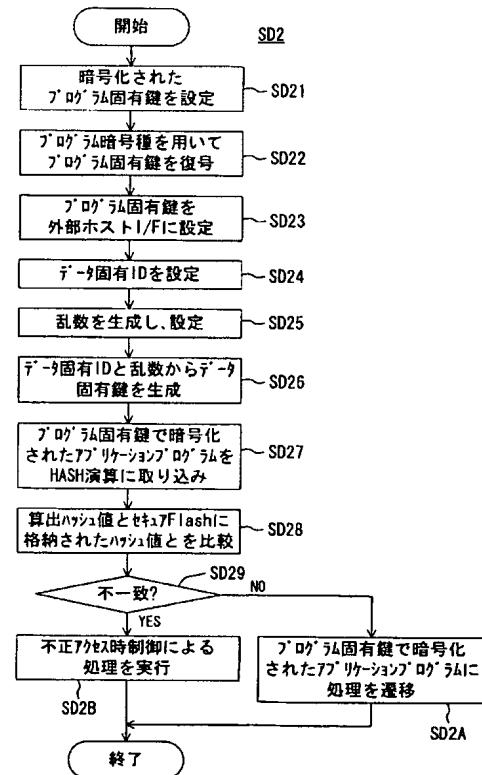
【図 17】



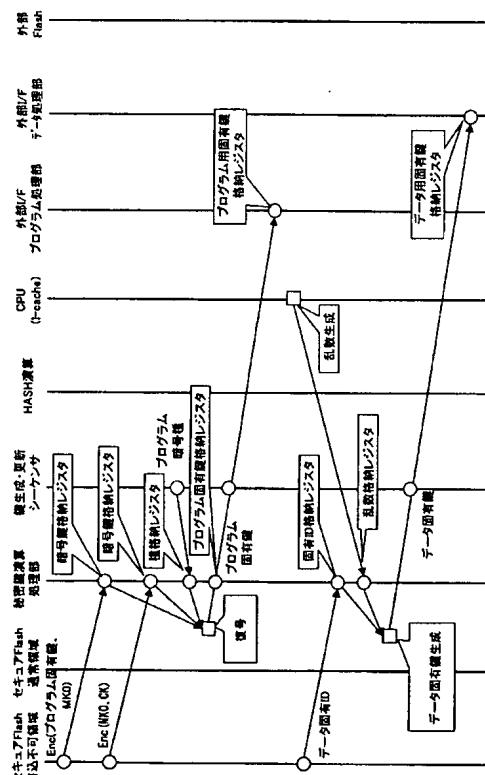
【図 1 8】



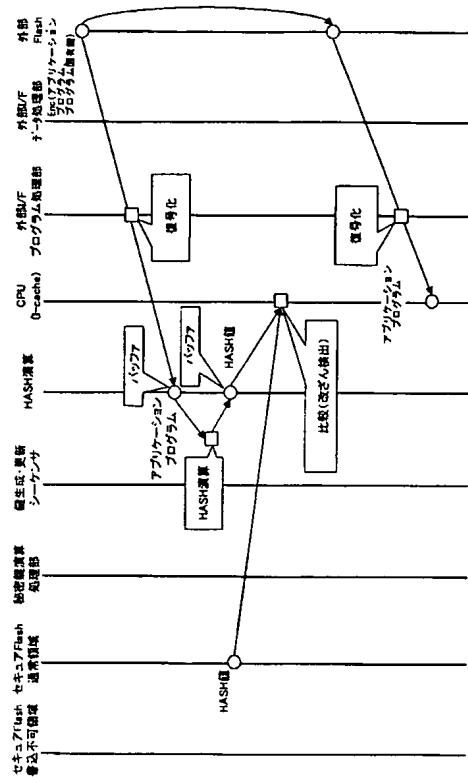
【図 19】



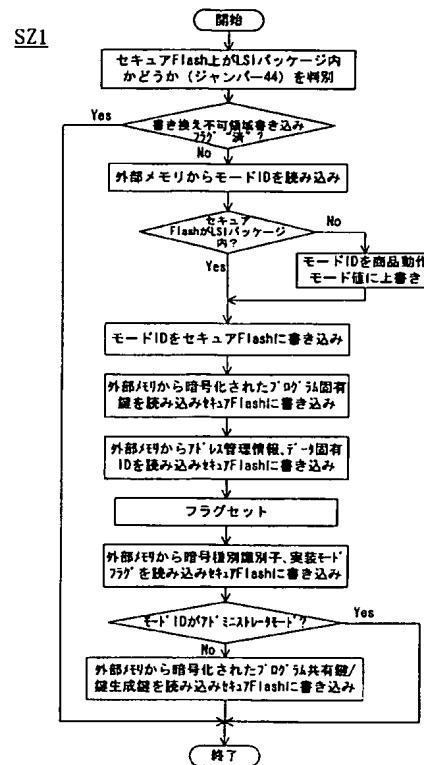
【图20】



【図21】



【図22】



---

フロントページの続き

(74)代理人 100115510  
弁理士 手島 勝

(74)代理人 100115691  
弁理士 藤田 篤史

(72)発明者 藤原 瞳  
大阪府門真市大字門真1006番地 松下電器産業株式会社内

(72)発明者 根本 祐輔  
大阪府門真市大字門真1006番地 松下電器産業株式会社内

(72)発明者 安井 純一  
大阪府門真市大字門真1006番地 松下電器産業株式会社内

(72)発明者 前田 卓治  
大阪府門真市大字門真1006番地 松下電器産業株式会社内

(72)発明者 伊藤 孝幸  
大阪府門真市大字門真1006番地 松下電器産業株式会社内

(72)発明者 山田 泰司  
大阪府門真市大字門真1006番地 松下電器産業株式会社内

(72)発明者 井上 信治  
大阪府門真市大字門真1006番地 松下電器産業株式会社内

F ターム(参考) 5B076 FA01  
5J104 AA12 PA14